# Dynamic Concept Drift Detection for Spam Email Filtering

L. Nosrati[1], A. Nemaney Pour[2]
[1]Sharif University of Technology, International Campus/Dept. of IT Engineering, Kish Island, IRAN
Email: nosrati.leili@gmail.com
[2]J-TECH Corporation/R&D Section, Yokohama, JAPAN
Email: a.nemaney@jtechno.jp

***Abstract*— Nowadays most of Internet users surfer from spam emails. Filtering technique is one of the effective methods which help us to get rid of the spam emails. One of the problems of filtering is that it cannot detect spam emails accurately when the concepts change or drift happens as time goes by. Therefore, it is required to handle concept drift accurately and quickly. This paper proposes a new algorithm for concept drift detection with three different levels; control, warning, and alarm level. The results show that the proposed algorithm can detect concept drift more accurately compared with the previously proposed ones. In addition, it can detect sudden concept changes more accurately.**

***Index Terms*— concept drift, content based filtering, machine learning, spam email, spam email detection**

## I. INTRODUCTION

Email is the most commonly-used forms of communication. It offers users' universal availability. However, increasing spam emails are serious threat for the email system. Spam email is a kind of unwanted email that is sent from one who is unknown to a massive number of individuals with various purposes in mind. Spam emails cause security problems. Most of Spam emails contain Trojans, Malwares, and viruses. In addition, most of spam emails are about pornography without correlation between receivers' area of interests and the contents. Finally, the cost of getting rid of spam emails is the third problem [1]. There are many filtering techniques to detect and to stop the flow of spam emails. Filtering techniques are catogorized into two groups, rule based filtering [2] and content based filtering [3]. Rule based filtering works through some certain rules and regulations by which the filter decides to pass or to block the received email. The problem of the rule based filtering is that the rules and the policies need to be updated by the administrator of the system continuously. This work is not an efficient and accurate work. On the other hand, content based filtering introduces machine learning which needs to be trained. The problem of this filtering is that the spammers are aware of its functionality and may use special characters to pass the filter. Because of the restrictions of the rule based filtering, we introduce content based filtering for our proposal. Concept change is one of the problems of filtering. Concept drift means that the concepts of the received emails change as time goes by. In other words, concept drift monitors the changes and the related implications in order to learn those changes. It may happen gradually or suddenly. Most of the filtering techniques cannot detect spam emails accurately when the concepts drift happens. Therefore, it is required to have a filtering system to handle concept drift accurately and quickly. This paper presents the algorithm of Dynamic Concept Drift Detection (DCDD). The purpose of this algorithm is to improve the accuracy and the performance of the previously proposed algorithms. This algorithm has three levels for its filtering; control level, warning level, and alarm level. The simulation results show that DCDD can detect concept drift more accurately and quickly compared with the previously proposed algorithms. The rest of this paper is organized as follows: section II introduces related work. In section III we describe the proposed algorithm for concept drift. Evaluation and experimental results are discussed in section IV. Finally, we conclude the paper in section V.

## II. RELATED WORK

In this section, we review the algorithms related to concept drift. Authors in [1] classify the concept drift algorithms in details. STAGGER was the first system which addressed concept drift. This system uses a distributed concept description comprised of class nodes interrelated to attribute-value nodes through probabilistic arcs. Concept versioning [1] is another concept drift system designed to cope with continuing evolutionary concept drifting. This system takes benefit from a frame representation, and manages such drifts by two methods; by altering current concept descriptions, or by making and creating a more recent version of these descriptions. The FLORA systems track concept drift by maintaining a sequence of examples over a dynamically adjusted window of time. It uses such examples to induce and refine three sets of rules; the rules covering the positive examples, the rules covering the negative examples, and the potential rules that are too general at present [4]. Ref. [1] claims that meta-learning mechanisms can recognize contextual features. This can be achieved by analyzing the frequency and occurrence of a learner's entire history as well as a fixed window of time. FLORA [5] and AQ-PM [6] family with differences have this capability respectively. Starting from FLORA2, the algorithm can store the most recently encountered examples over a dynamically sized period of time. FLORA3 has mechanisms for coping with noise. Flora4 has extensions to deal with recurring contexts. On the other hand, AQ-PM uses the AQ algorithm. The system learns new rules from those ones stored in memory, and from new ones in the input stream. Those rules are not used after a fixed


DOI: 01.IJNS.02.04.64

period of time. Although, AQ11-PM stores boundary examples, it forgets them after a fixed period of time. It uses the AQ11 algorithm to form concepts incrementally. All of these systems have been evaluated on STAGGER concepts.Ref. [1] explains online algorithms for training support vector machines. The special feature of this algorithm is that it adds the formerly obtained vectors to the recent training set, and builds a new different machine. On the other hand, instance selection, weighting, and ensemble learning [7, 8] are three most common measures taken to manage the effects concept drift. Building upon instance selection and involving generalization from a window are the most common ways to handle concept drift. These algorithms are called learning with multiple concept descriptions. Finally, Drift Detection Method (DDM) [9], and Early Drift Detection Method (EDDM) [10] are two typical concept drift algorithms in this research area. While DDM shows good behavior for sudden change detection, it has difficulties when the changes happen slowly and gradually. On the other hand, EDDM improves the gradual changes detection.

## III. Proposed Algorithm

Generally, an email consists of a header, and the body. The header includes the electronic addresses of the sender and the recipient(s) and the subject of the email. The body contains all the information that the email is composed for in html or text format. The following preprocessing steps are applied to the algorithm of DCDD.

### A. Preprocessing system

1) Email sort: emails are sorted by time assuming that the vocabulary of each e-mail is similar.
2) Tokenization: The words in the e-mails are separated from each other, and each word is considered as a token.
3) Stop word removal: Stop words such as “to”, “a”, and are deleted.
4) Html removal: All the html tags are cleared.
5) Attachment removal: All the attachments are removed and instead we write <attachment>.
6) Lemmatization: All words are returned to their root. For example the word “receiving” is changed to “receive”.
7) Representation: The words are changed to the usable forms for algorithms. In other words, text classification is performed.
8) Feature selection: Features of all emails are extracted and saved in feature vector. By feature selection, the measure of efficiency is increased. Hence, feature selection has important role on reducing the dimension of the vectors.

| Parameters | Description |
|---|---|
| $n$ | the number of examples learned by classifier |
| $s$ | the number of correct classifications among most recent $W$ examples |
| $r$ | the number of correct classifications among $n$-$W$ examples |
| $W$ | window size |
| $\alpha$ | one-tailed significance level |
| $a_W$ | drift level |
| $a_d$ | warning level |
| $P$ | $P$-value test |
| $P_A$ | overall accuracy |
| $P_B$ | the recent accuracy |
| $r_i$ | the number of correct classifications among $n$-$W$ examples from $i$ examples |

### B. The Algorithm

Before we proceed further, we introduce the notation used in this paper as shown in Table I. The algorithm consists of three different levels; control level, warning level, and alarm level. In control level no changes happen. When the algorithm reaches the warning level, it thinks about the changes that may happen. At last, when the probability of warning level increases, then the algorithm shifts to alarm level. In this level, concept drift happens. The following explains how the algorithm works.

### A. The Algorithm

Before we proceed further, we introduce the notation used in this paper as shown in Table I. The algorithm consists of three different levels; control level, warning level, and alarm level. In control level no changes happen. When the algorithm reaches the warning level, it thinks about the changes that may happen. At last, when the probability of warning level increases, then the algorithm shifts to alarm level. In this level, concept drift happens. The following explains how the algorithm works.

- DCDD stores training examples in short-term memory.
- This memory slightly improves the predictive accuracy immediately after the rebuild of the online classifier.
- First $W$ is defined as the length of window size.
- Data are classified by the base learner algorithm.
- Results feed to the proposed algorithm.
- This algorithm uses two levels of significance $a_W$ and $a_d$.
- The algorithm checks the data in the window to find how many of them have been classified correctly. Then, it compares them with the data which are not in the window.
- While $r/(n - W) > s/W$ and $P < a_d$, the algorithm is reached the warning level.
- In warning level, the algorithm rebuilds the online classifier from the stored examples. $n \geq 2W$.
- The input data is the stream which is the set of vectors. Each vector is the instances with their features.
- The concept drift detection starts working when
- The algorithm compares two rates $P_A$ and $P_B$ by the statistical method. If the results are very different from each other, it shows that concept drift has happened. When $P_A = P_B$ concept drift does not occur.

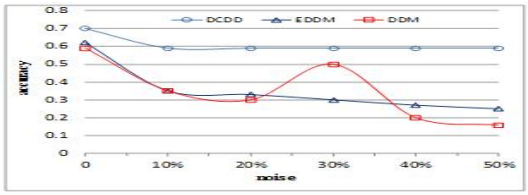


Figure 1. Comparison of accuracy of DCDD, EDDM, and DDM in front of noise.

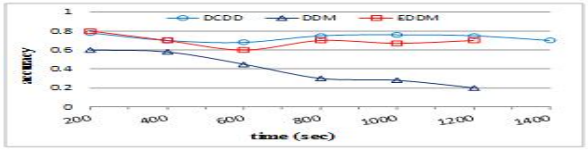


Figure 2. Comparison of accuracy of DCDD, EDDM, and DDM, in front of sudden changes.


DOI: 01.IJNS.02.04.64

•Finally, All the variables are reinitialized when $r/(n - W) > s / W$ and $P < a_w$

## IV. Comparision

In this section, we analyze and compare DCDD with two typical algorithms, EDDM and DDM by simulation results. The recent accuracy is obtained by $P_B = r_B / n_B = s / W$. Again, the overall accuracy is obtained by $P_A = r_A / n_A = r / (n - W)$. Figs. 1 and 2 summarize the comparisons focussing on the following measures.

•Noise: feature selection with low degree of changes during long period of time.

•Sudden changes: feature selection with high degree of changes during short period of time.

•Kappa statistics: the consistency between predicted and the true value.

•Recall: The fraction of all spam messages classified by the filter to be spam.

•Precision: The fraction of messages classified by the filter as spam that actually are spam.

•F-measure: The value of F measure can show how accurate the algorithm has been.

TABLE II. **Comparison of Accuracy Rates**

| Error & Accuracy | DCDD | EDDM | DDM |
|---|---|---|---|
| Precision | 0.776 | 0.537 | 0.501 |
| Recall | 0.791 | 0.555 | 0.512 |
| F-Measure | 0.777 | 0.545 | 0.521 |

Fig. 1 illustrates the comparison of simulation results in front of noise. We started the noise with 10% and increased it to 50%, and calculated the accuracy. Before applying the noise, the accuracy of DCDD method was 0.7. After applying 10% to 20% noise, the accuracy decreased to about 0.6. However, after applying 20% noise to the algorithm, the system got stable, In addition, the simulation results with EDDM and DDM show that the accuracy of these methods is about 0.62 without noise. However, after applying 20% noise to the system, the accuracy of EDDM suddenly decreases to 0.35, and when the noise reaches 50%, the accuracy decreases sharply to 0.25. On the other hand, DDM does not show any stable state when the noise increases. As a result, the accuracy of DCDD is steady in front of noise compared with EDDM and DDM. Fig. 2 illustrates the comparison of simulation results in front of sudden changes. As shown, the accuracy of DDM decreases in front of the sudden changes. Although the accuracy of EDDM is better than DDM, DCDD shows higher accuracy compared with EDDM. As a result, during the simulation the accuracy of DCDD was approximately stable in front of sudden changes. Kappa statistics or performance shows the consistency between predicted and the true value. The higher consistency between predicted value and the true one causes higher kappa statistics.

The result shows that the Kappa statistics of DCDD is 0.38 compared with EEDM and DDM with is 0.04 values. This shows that the performance of the proposed method is higher. Table II shows the comparison of accuracy rates for DCDD, EDDM, and DDM. The results show that all the accuracy rates for our proposed model are higher than the previously proposed ones.

## V. Conclusion

In this paper, spam problems, spam filtering, and concept drift were presented. In addition, we proposed the algorithm of Dynamic Concept Drift Detection (DCDD). We compared the simulation results of DCDD with two typical algorithms, EDDM and DDM. At the end, we conclude our proposal with some of its contributions:

- DCDD can detect sudden changes more accurately.
- DCDD is approximately steady to face with noise.
- DCDD has high kappa statistics or performance.
- DCDD has improved the accuracy rates such as precision, recall, f-measure.

## References

[1] T. S. Guzella, and W. M. Caminhas, "A Review of Machine Learning Approaches to Spam Filtering," *Elsevier, Expert Systems with Applications*, vol. 36, no. 7, pp. 10206-10222, 2009.

[2] ýA. Ciltik, and T. Gungor, "Time-Efficient Spam E-mail Filtering using n-Gram Models," *Pattern Recognition Letters*, vol. 29, no. ýý1, pp. 19-33, 2008.ý

[3] E. Blanzieri, and A. Bryl, "A Survey of Learning-based Techniques of Email Spam Filtering," *Artificial Intelligence Review*, vol. 29, no.1, pp. 63-92, 2008ý

[4] I. Zliobate, "Learning under Concept Drift: an Overview," *Technical Report on Artificial Intelligence*, Vilinios University, pp. 371–391, 2010.

[5] Q. Zhu, X. Hu, Y. Zhang, and P. Li, "A Double-Window-based Classification Algorithm for Concept Drifting Data Streams," *proceedings of IEEE International Conference on Granular Computing (GrC)*, CA, USA, 2010, pp. 639-644.

[6] Z. Ouyang, and M. Zou, "Mining Concept-Drifting and Noisy Data Streams using Ensemble Classifiers," *proceedings of IEEE International Conference on Artificial Intelligence and Computational Intelligence (AICI 2009)*, Shanghai, China, 2009, pp. 360-364.

[7] A. Tsymbal, "The Problem of Concept Drift: Definitions and Related Work," *Technical report TCD-CS-2004-15*, Trinity College Dublin, Ireland, pp.123-. 130, 2004.

[8] J.Z. Kolter, and M.A. Maloof, "Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift," *Proceedings of IEEE Third International Conference on Data Mining*, Washington DC, USA, 2003, pp. 123-130.

[9] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with Drift Detection," *Lecture Notes in Computer Science*, vol. 3171/2204, pp. 66-112, 2004.

[10] M.B. Jose, J.D.C. Avila, R. Fidalgo, A. Bifet, R. Gavalda, and R.M. Bueno, "Early Drift Detection Method," *Fourth International Workshop on Knowledge Discovery from Data Streams*, Berlin, Germany, 2006, pp. 77-86.




DOI: 01.IJNS.02.04. 64